Adobe® LiveCycle® Enterprise Suite

Contents

Introduction

As Enterprises face the dual challenges of making efficient use of back-end resources while more effectively engaging with customers beyond the firewall, IT departments need to make investment choices that allow them to better organize and use distributed capabilities. These capabilities must work in concert with scalable applications that engage customers through the appropriate and targeted use of rich and compelling interfaces.

Meeting the challenges of effectively reaching and engaging customers against a backdrop of rapidly changing business environments, while reducing development costs and maximizing return on investment, demands a service-oriented approach to application delivery that allows back-end processes to be exposed as loosely coupled services and easily assembled into targeted solutions. These applications must be presented to customers through rich and intuitive interfaces that increase user satisfaction, improve process utilization and completion rates, and increase transaction size.

This paper discusses the Adobe® LiveCycle® Enterprise Suite (ES) architecture, providing an overview of how LiveCycle ES employs service-oriented principles that allow faster development utilizing an integrated IDE and increases confidence with security technology such as rights management, encryption, digital signatures and auditing. The paper also explores how organizations can quickly develop engagement applications and improve the effectiveness of existing tools, blending PDF and Flex technologies to provide users with uniquely engaging applications.

LiveCycle ES exposes software as services and enables effective customer engagement applications that meet these challenges by allowing organizations to quickly assemble secure, feature-rich solutions that leverage Adobe® PDF and Flex™ technologies to dramatically improve user engagement via the cross-platform Adobe® Reader® and Adobe® Flash® software installed on over 700 million devices worldwide. Organizations can choose the appropriate combinations of standards-based user interface technologies, including Flex, PDF and XML, HTML, and AJAX, to most effectively engage their users, drive flexibility, and reduce custom implementations and vendor lock-in.

LiveCycle ES Architecture Overview

The LiveCycle ES architecture encompasses clients, user interface technologies, development tools, and foundation infrastructure. The following sections discuss these themes in the contexts of designing, running, integrating, and administering customer engagement applications.

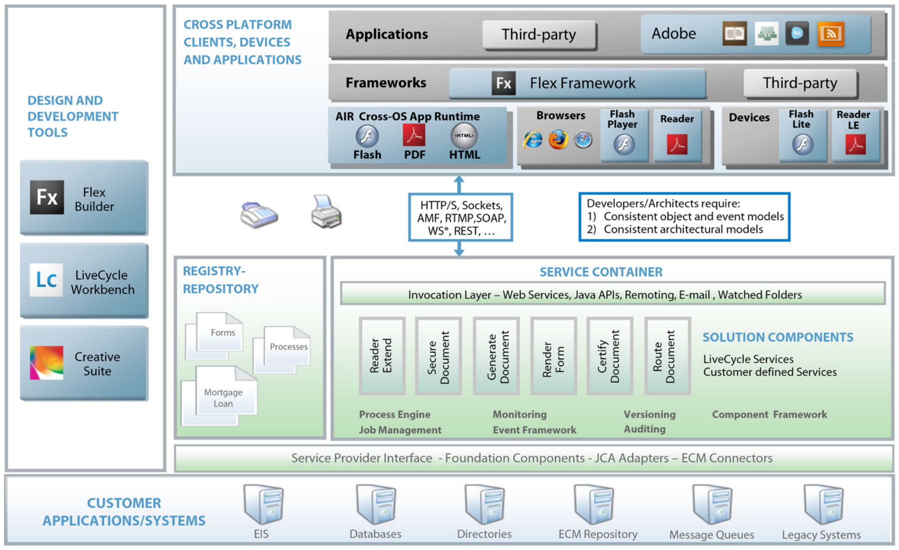


Figure 1: The LiveCycle ES architecture

The LiveCycle ES design and development tools work within the application model to coordinate shared assets and dependencies, streamlining the development process and allowing multiple users to work on the same application simultaneously. Design-time revision control, security, and auditing components provide administrators with the tools they need to ensure end-to-end management of the application development process.

LiveCycle ES Solution Components provide reusable services that can be easily assembled into customer engagement applications. These components provide wide-ranging and robust support for data capture and business transformation applications.

The LiveCycle ES Foundation provides an integrated environment that enables the deployment, execution, and management of customer engagement applications. Integral to the Foundation is the Service Container, which provides a common runtime environment for all solution components and associated services. The LiveCycle ES Foundation services provide facilities that enable solutions to integrate with common IT infrastructure, such as providing coordinated encryption services to orchestrated components within the service container.

LiveCycle ES includes centralized tools that simplify a wide range of administrative tasks, including configuration, administration, deployment, and monitoring. Each solution component can be independently administered from a single portal location.

LiveCycle ES Architecture In-Depth

Designing Engagement Applications

LiveCycle Workbench ES and Flex Builder provide integrated form, process, and rich Internet application design tools that expose the appropriate functionality to specific team members. For example, form designers and business analysts can work collaboratively within the form perspective of LiveCycle Workbench ES, while developers can extend application functionality through data integration, scripting, and extension of component capabilities through the LiveCycle ES SDK. Business analysts can define processes while form designers can reuse form fragments and script components created by designers. Reusable content components (e.g., form templates, service definitions, process definitions, policies, DDX files, XML schemas, and images) and form fragments (e.g., headers and footers that can be reused across multiple forms) are stored, version-controlled, and secured in LiveCycle ES. This approach greatly simplifies the maintenance and update tasks related to application development. Repositories can be seamlessly extended with Enterprise Content Management (ECM) connectors, and assets can be secured using access control lists.

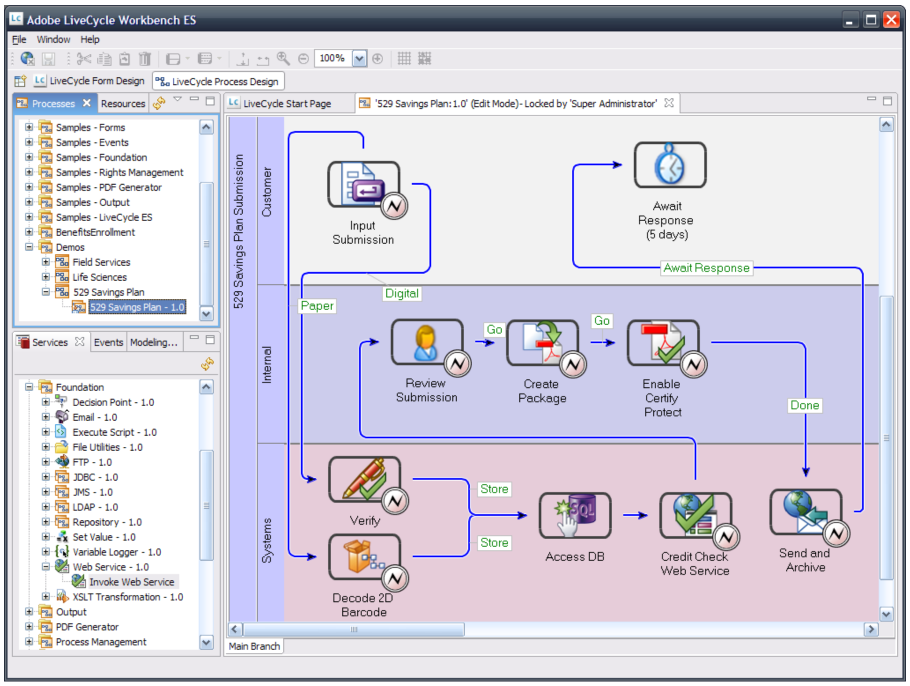


Figure 2: The LiveCycle Workbench ES Process Design view simplifies business process modeling with concepts such as swim lanes

LiveCycle Workbench ES unifies authoring tasks for forms, processes, templates, and components. Integration with the repository simplifies management of artifacts, providing support for critical management activities, such as revision control, security, and auditing. The process design perspective gives business analysts the ability to create processes using standard Business Process Modeling Notation (BPMN) constructs, for example, swim lanes and pools. The form design perspective embeds the LiveCycle Designer ES functionality for the graphical layout of XML form templates that can be rendered as PDF, Flash, or HTML by LiveCycle Forms ES. Forms are easily integrated with back-end data sources and destinations to build applications that automate common tasks, such as data capture. Users with existing XForms can leverage that investment using import facilities to render XForms in HTML, PDF, and Flash formats. Based on Eclipse, LiveCycle Workbench ES minimizes up-front training, costs, and risks, and supports the Eclipse plug-in environment.

LiveCycle ES provides the broadest array of clients, formats, and design tools for user interaction and data capture and supports development of RIAs using Flex Builder and the underlying Flex Framework. Flex technology leverages the Flash client, enabling organizations to create desktop-like solutions that reach more than 96% of Internet-connected desktops. Client-side ActionScript coupled with data access through LiveCycle Data Services ES and common mechanisms like web services and HTTP/REST simplify the development of applications, such as business decision support, customer support, and data capture. The Flex Framework provides more than 100 interface components. Flex user interfaces are described in industry standard MXML. Since Flex support is fully integrated with LiveCycle ES, developers can take advantage of platform services to manage and deploy Flex components. Form designers can also utilize Form Guides as part of the form development process to help users step through a task in a logical sequence. Form Guides simplify data entry for users by breaking the process of completing a form down into a sequence of smaller, thematically linked activities presented in a rich Flash interface. Form Guides can be created from existing forms with an intuitive Guide Builder without the need for additional code.

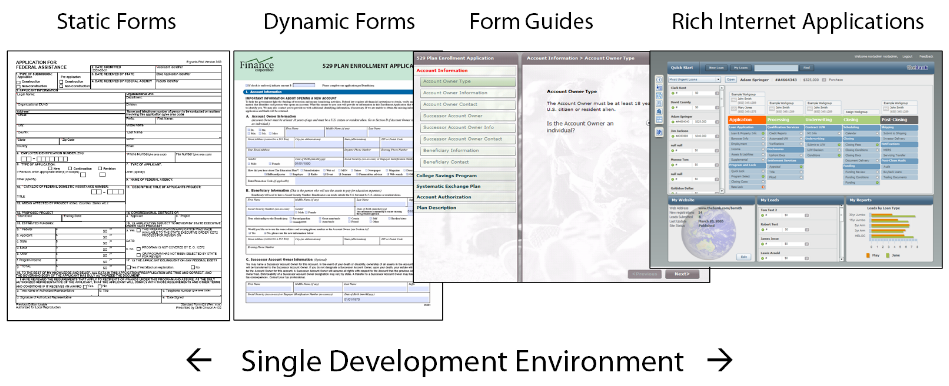

Static Forms
Dynamic Forms
Form Guides
Rich Internet Applications
← Single Development Environment →

Figure 3: LiveCycle ES allows appropriate interfaces to be created with a common forms and data model to improve flexibility and engagement

While the request/response communication model is sufficient for website browsing, many applications require optimized high-performance data transfer as well as additional modes of interaction, including publish/subscribe messaging and the ability to push data or alerts from the server to the client. The LiveCycle Data Services ES component enables high-volume data exchange and synchronization between Flex- or AJAX-based application interfaces and back-end systems.

Component Versioning

Many production environments need to maintain backward component compatibility for existing applications while leveraging the bug fixes and additional features found in newer versions. Historically, these competing needs have demanded compromises and fragile work-arounds. LiveCycle ES provides designers and developers with integrated version management, allowing multiple versions of the same components to coexist and function without conflict within the same environment. Any number of versions of a component may be deployed under the same service binding. LiveCycle ES uses developer-supplied metadata to establish policies on how to manage dependencies between components and their clients.

Client developers have the following choices when requesting interaction from a LiveCycle ES service:

- No version specification: All requests will be routed to the latest version of the service.
- Explicit specification of the original version: If the original binding version is specified, all requests can be routed to that version or compatible newer version.
- Specific version request: Client requests will be routed to only the version specified.

Deploying

LiveCycle ES application commissioning tasks related to testing, staging, and deploying to production servers are simplified through the use of LiveCycle Archive (LCA) files that automatically package application-related resources in a single archive for transfer between individuals and systems. Administrators can stage or deploy applications to production servers using LiveCycle administrative tools. No development infrastructure is required on the servers to which applications are deployed.

Running Applications

The LiveCycle ES architecture embraces Service Oriented Architecture (SOA) principles that enhance an organization's ability to organize and use distributed capabilities. As described in this paper, LiveCycle ES provides a framework in which business analysts, application architects, and developers can match business needs with capabilities by exposing back-end processes as services that are readily assembled into targeted solutions, greatly improving an organization's agility to meet changing business needs.

LiveCycle ES exposes business functionality as discrete services with clearly defined interfaces that allow them to be re-purposed and used as required by other applications. For example, LiveCycle Rights Management ES provides the same robust, document-level access control to all applications requiring this functionality. Since exposed services will often operate across domains of ownership, organizations must be able to declare policies that describe a service in technical and business terms, encompassing security, governance, and related functions. LiveCycle ES services are deployed with XML descriptions that qualify their use.

The following sections describe key aspects of the LiveCycle ES runtime functionality.

Invocation Layer

The LiveCycle ES invocation layer allows people and external applications to initiate processes via a wide range of mechanisms. The invocation layer intercepts invocation requests, performs security, auditing, logging, and transaction management functions and ultimately invokes a target service. Any client, such as .NET, PHP, or e-mail can start a service. This unified invocation framework reduces complexity and improves interoperability.

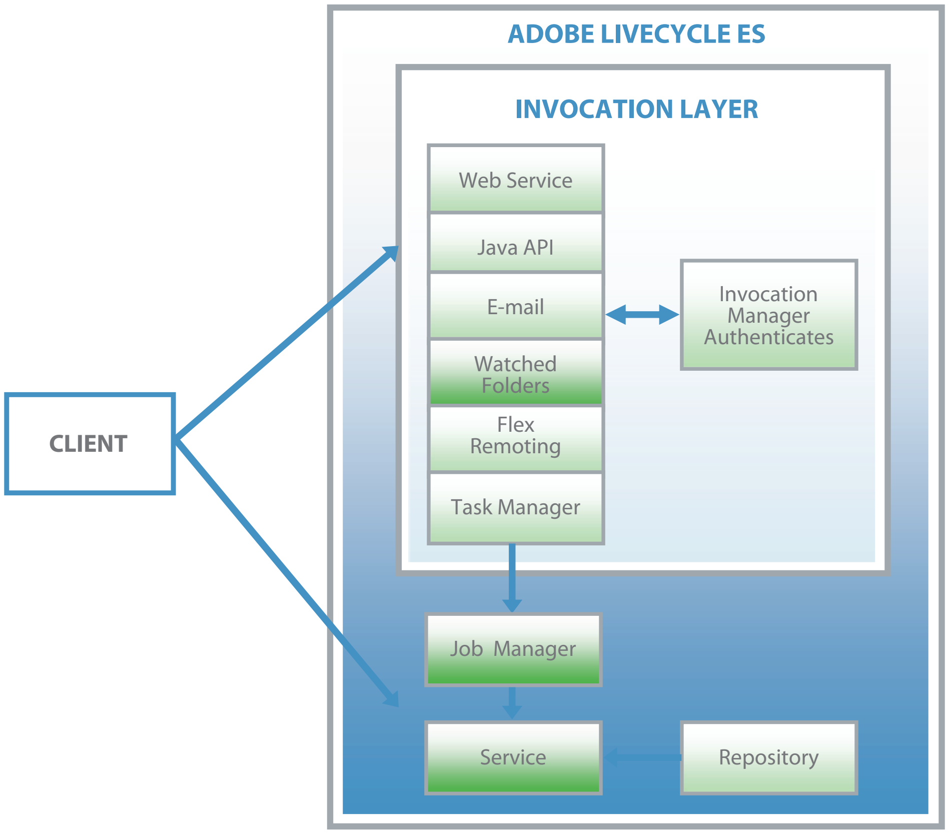


Figure 4: Flow of activity when a client makes an asynchronous request to a LiveCycle ES service. Synchronous communication is also available.

LiveCycle ES provides several integration methods for invoking a service:

- **Web Services**: Service invocation via Web services is Web Services Basic Profile 1.1 compliant. It supports SOAP with MIME and DIME attachments, the WS Security 1.0 standard, dynamically generating a SOAP endpoint and WSDL. Web service invocation allows both synchronous (short-lived) and asynchronous (long-lived) service requests.
- **Java API**: Service invocation via Java utilizes Remote Method Invocation (RMI) and utilizes a strongly typed API. Java-based invocation methods are highly optimized for document throughput and provide excellent performance when submitting a large document to a service.

- **E-mail**: A service can be invoked when a specified e-mail account receives an e-mail. This invocation method supports IMAP, POP3, and SMTP using both basic authentication and SSL. Designers can define patterns for e-mail attachments and map data from e-mail headers, body, and attachments. For example, a PDF e-mailed to a specified account could invoke an "apply policy" operation and return the PDF to the sending e-mail address.
- **Watched Folders:** Services can be invoked by writing files to network folders that LiveCycle ES has been configured to watch. The folders are scanned using configurable filename patterns such as .pdf or sales.* and a desired processing action taken. Processed output files can be written back to a designated folder for consumption by external applications or users.
- **Flex Remoting**: Flex clients can call LiveCycle ES services via the Flex RemoteObject tag. This method provides ActionScript to LiveCycle ES bridging, supports synchronous and asynchronous invocations, and provides dynamic endpoint creation, saving substantial development resources.
- **Task Manager**: By exposing a service as a task in LiveCycle Workspace ES, users with appropriate access rights can manually invoke that task from the Workspace Start Processes list. For example, an employee returning from a business trip could start an expense submission process.

After an asynchronous request is received via one of these invocation methods, the message receiver authenticates the request and creates a normalized invocation request. It looks up the service and operation name and passes this information to the router.

Foundation

The LiveCycle ES foundation provides underlying functionality common to all applications, including routing and management of requests from clients. When an asynchronous request's service and operation details are retrieved, the router calls the Invocation manager. A JobId is then returned to the client through the receiver.

Job Manager

Operations performed by services in LiveCycle ES can be either short-lived or long-lived. Short-lived operations complete synchronously on the same thread from which they were invoked. Familiar from most programming languages, these operations block while awaiting a response.

However, many tasks performed in Enterprise workflows cannot be expected to complete synchronously. These long-lived operations often span systems or even extend beyond the organization, as when a client must complete and submit a loan application form as part of a larger solution that integrates multiple automated and human tasks. Such operations must not block while awaiting a response. Long-lived operations perform their underlying work asynchronously, permitting resources to be otherwise engaged while awaiting completion. Unlike a short-lived operation, the LiveCycle ES job manager does not consider a long-lived operation complete once it is invoked. An external trigger, such as a system requesting another operation on the same service or a user submitting a form, must occur to complete the operation. The LiveCycle ES job manager utilizes industry-standard Java Messaging Services (JMS) to receive status information. LiveCycle ES provides a ready-to-use framework for long-lived operations, including tracking and recovery.

Service Container

The LiveCycle ES Service Container is a single, scalable, and unified runtime environment based on SOA principles, in which all standard and custom LiveCycle services execute. The Service Container functionality includes orchestration of services into short- and long-lived processes, monitoring, auditing, and coordinated security. Developers and administrators also benefit from simplified and fine-grained development, deployment, securing, and maintenance of services. When an asynchronous client request has been authenticated, it is passed to the Job Manager, which invokes the service.

Several key attributes of the Service Container are described below:

- **Component Model**: The Service Container employs an extensible component model in which loosely coupled services can interact to provide compelling customer engagement experiences. LiveCycle ES Foundation provides services that enable the LiveCycle ES platform to integrate with common enterprise infrastructure such as directory servers over LDAP, Web services, JDBC, FTP, and file systems. LiveCycle ES also encompasses standard solution services such as LiveCycle Reader Extension ES for to enable enhanced features in Adobe Reader®, and LiveCycle Data Services ES to integrate customer engagement applications with LiveCycle services and back-end systems.

- **Orchestration**: A key aspect of the LiveCycle ES runtime environment is a set of facilities that orchestrate processes to complete complex tasks. Organizations can orchestrate multiple processes together into a new service, provide end-user interaction with LiveCycle ES services as part of long-lived business processes, and integrate services with third-party products and services.

  By orchestrating existing document services, solutions can be easily assembled and tailored to specific requirements. For example, consider a requirement to retrieve a PostScript file from disk, convert it to PDF format, and save it back to another disk location. Developers can easily meet this requirement using LiveCycle Workbench ES to combine three existing services into a single, orchestrated service called ConvertPS. This service uses a single invoke operation and takes full advantage of the invocation methods available to LiveCycle ES.

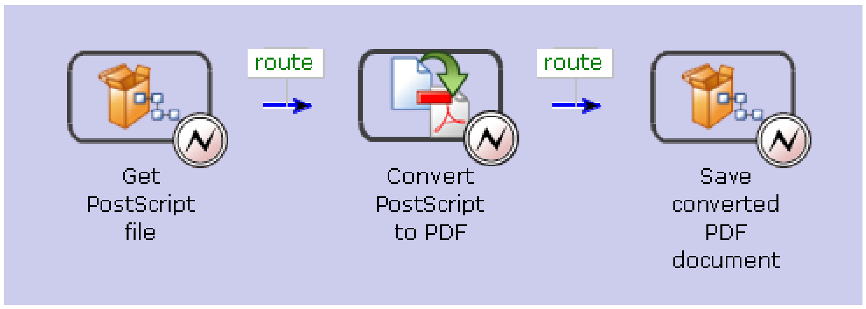
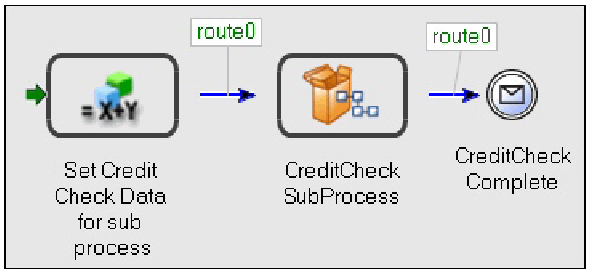


Figure 5: The ConvertPS orchestrated service

- **Event Framework**: LiveCycle ES provides a framework through which solution developers can raise and receive customized events to facilitate communication between running processes. They can be used to accomplish tasks such as initiating, terminating, or causing a state change in a process when a dependant operation in a related process completes. Information about an event is stored in an XML variable available to the event receiver. Events can be filtered to determine which event filters are triggered. LiveCycle ES supports asynchronous, exception, and timed events.



A representative process utilizing the event framework is a loan application in which a Credit-CheckComplete asynchronous event is thrown after the credit check completes. This process is initiated when the customer submits a loan application.

Figure 6: The loan application process uses the Event Framework to asynchronously signal completion of a credit check

The CreditCheckComplete event receiver functions as the starting point for the subsequent ApproveApplicant process.

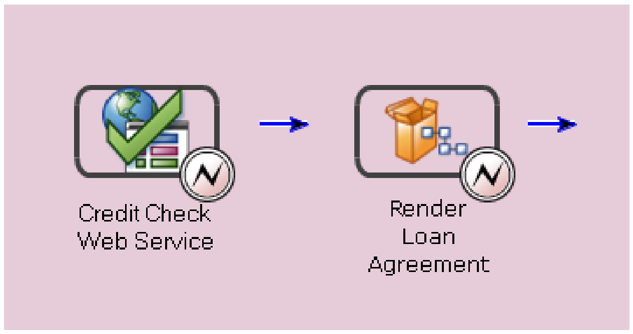


Figure 7: Upon completion of the credit check, additional processes may be invoked automating additional business activities

Registry and Repository

The LiveCycle ES repository provides application developers with advanced facilities that greatly enhance the accessibility and management of assets. The repository seamlessly spans a single or multiple clustered J2EE systems, simplifying scalable and failover deployments. Versioning of deployment assets helps structure and control these activities at runtime. Components and services in the central repository are inventoried in respective registries that facilitate browsing, lookup, starting, stopping components and services, and deploying and undeploying services.

Adobe® LiveCycle® Connectors for ECM provide simple integration between LiveCycle ES and Enterprise Content Management systems such as EMC Documentum® and IBM Filenet®.

When a service is invoked, the registry and repository are utilized to access runtime assets required to complete the request. Clients poll the status of their asynchronous requests and when the job is marked as complete, they call the service manager to retrieve the result.

Integrating Applications in the Enterprise

LiveCycle ES Foundation provides a comprehensive range of methods for integrating both as a server and as a client of other processes.

In addition to the invocation methods described above, LiveCycle ES supports several methods for requesting the services of other processes in the enterprise and returning the result to an application. These methods provide flexibility to leverage existing resources.

- **Web services**: A service can also be implemented as a proxy to an existing web service. The proxy definition in the component.xml file contains the information necessary to define the operations and parameters to be exposed as part of the proxy service.
- **Custom components**: Developers can add their own services by creating stateless plain old Java objects (POJOs) containing one or more operations. Java developers can often reuse existing code with minimal effort. The POJO, a component.xml file describing it, and any dependencies, for example, libraries, are combined into a single JAR file. The document service is then deployed to the LiveCycle Service Container using familiar tools and steps as in any other deployment.
- **Connectors for ECM**: Connectors provide content repository services such as check in/out, lock, and content meta data access. The connector service components receive and output document and other content objects through LiveCycle process, connecting ECM to customer applications.
- **JMS**: LiveCycle ES supports the JMS (Java Messaging Services) API for communicating with other Java-based applications in a distributed environment.

- **SPI**: LiveCycle ES includes a service provider interface to standard external authentication methods, for example, Lightweight Directory Access Protocol (LDAP) servers, Microsoft® Active Directory®, and custom authentication methods.
- **Foundation services**: LiveCycle ES supports a number of out-of-box components and libraries that provide core functionality for integration with common IT infrastructures. Support includes integration to the user directory through LDAP, authentication, ECM, JMS, RMI, e-mail and other back-end systems, applications, and resources.
- **NetManage JCA Adaptors**: JCA Adapters allow easy connectivity to back-end systems, without requiring developers to learn the inner workings of those systems. JCA adapters expose the target systems' functions and transactions, making them available to composite applications, Enterprise Service Bus (ESB), and any middleware technology. JCA adapters facilitate bi-directional synchronous and asynchronous communication between applications, improving the effectiveness and value of existing systems and information.

  LiveCycle ES customers can leverage NetManage LCA Adapters with connectivity to more than 40 enterprise information systems including SAP, Oracle, Siebel, PeopleSoft, Tibco, IBM, and Lotus.
- The LiveCycle Data Services ES component provides real-time messaging, quality of service, and data management services for high-volume data exchange and synchronization between Flex or AJAX based application interfaces and back-end systems.

Administering Applications

The LiveCycle Administration Console is a web-accessible tool that system administrators use to perform a wide range of tasks, such as adjusting port number and logging options, and managing the deployment and configuration of LiveCycle ES applications. Administrators also use the LiveCycle Administration Console to create and manage service endpoints and to deploy applications. It also provides a common authentication management service used throughout the platform and across all services, easing tasks such as defining and configuring users and groups and integration with LDAPs. Administrators also have access to JMX compatible service monitoring via the LiveCycle Administration Console.

Administrators can also create LiveCycle Archive (LCA) files that contain an application's process definitions and related resources such as forms, images, and XML files. LCAs simplify the export and transfer of applications between systems, and can be imported into LiveCycle ES without stopping the system. This mechanism eases the workload for organizations wanting to stage solutions from development through testing and deployment, and simplifies the sharing and distribution of applications with partners and customers.

The LiveCycle Configuration Manager is also used during installation to configure the initial LiveCycle ES platform settings, deploy solution components, and perform maintenance tasks, for example, applying service packs.

The BAM Workbench allows system administrators to quickly set up business connectivity monitoring by configuring event and contextual data sources and accessing multiple, concurrent data streams. System administrators can also create business rules and dashboard objects, as well as views and cubes in the Workbench. Administrators can define criteria ranging from simple to complex to trigger alerts.

Conclusion

LiveCycle ES is a scalable, unified platform that blends electronic forms, process management, document security, and document generation to help Enterprises create and deliver rich and engaging applications that reduce paperwork, accelerate decision-making, and ensure regulatory compliance. LiveCycle ES provides the tools to develop and deliver applications in a single, J2EE compatible and standards-based environment that eases integration of LiveCycle ES applications with existing IT assets.

Organizations can use LiveCycle ES to solve diverse problems ranging from data capture to complex business transformations. For example, government agencies support processes ranging from electronic tax submissions to full eGrant application management. Financial services solutions include loan application processing and account opening that help institutions distinguish themselves in a highly commodatized market. Manufacturers use LiveCycle ES to collaborate securely on product design with external partners. Life Sciences organizations automate sensitive processes, for instance, patient information management, confidently by using security features of LiveCycle ES including encryption, rights management, and auditing to help ensure their compliance with regulatory frameworks, such as HIPAA.

LiveCycle ES eases development through an integrated toolset that enables business analysts, developers, form designers, and others to work collaboratively with shared, reusable resources. Central to LiveCycle ES is a single runtime environment that provides invocation, event management, process management, and other essential functions to Foundation and Solution components. Integration with users and external processes is facilitated through common invocation and request methods such as Flex remoting and web services.

The Service Oriented design of LiveCycle ES greatly simplifies the assembly, deployment, and maintenance of customer engagement applications, combining rich and effective user experiences with back-end processes exposed as services in a consistent way. By assembling customer engagement applications from loosely coupled services in the LiveCycle ES platform, organizations can reduce development, deployment and maintenance complexity, align IT systems with business priorities, reduce costs, and improve agility and return on investment.

Adobe®

Adobe Systems Incorporated
345 Park Avenue
San Jose, CA 95110-2704
USA
www.adobe.com